CLAIMS:

What is claimed is:

Sub A7

1. A method in a data processing system for debugging a process from a starting point, comprising:

initiating debugging of a process;

saving a process state in response to a first event to form a stored process state;

retrieving the stored process state in response to a second event; and

reinitiating debugging from the stored process state.

2. The method of claim 1, wherein the first event occurs periodically.

3. The method of claim 1, wherein the process state is saved in a checkpoint data structure.

4. The method of claim 3, wherein the checkpoint data structure is a checkpoint file.

5. The method of claim 3, wherein the checkpoint data structure includes a process descriptor for the process.

6. The method of claim 5, wherein the process has control over at least one child process and the checkpoint data structure includes a process descriptor for each of the at least one child process.

7. The method of claim 5, wherein the checkpoint data

Sub A'7

structure further includes at least one data type descriptor for at least one data type corresponding to the process.

8. The method of claim 7, wherein the checkpoint data structure further includes at least one instance descriptor for at least one instance of data corresponding to each of the at least one data type.

9. The method of claim 8, wherein the checkpoint data structure further includes at least one data block corresponding to each of the at least one instance descriptor.

10. The method of claim 1, further comprising the step of modifying at least one register or memory variable before resuming debugging from the stored process state.

11. The method of claim 1, wherein the process state is saved when the program is in a stopped state.

12. The method of claim 11, wherein the stopped state is at a breakpoint.

13. An apparatus for debugging a process from a starting point, comprising:

a processor; and

a memory electrically connected to the processor, the memory having stored therein a program to be executed on the processor for performing the following steps:

initiating debugging of a process;

Sub A'7

saving a process state in response to a first event to form a saved process state;

retrieving the saved process state in response to a second event; and

reinitiating debugging from the saved process state.

14. The apparatus of claim 13, wherein the first event occurs periodically.

15. The apparatus of claim 13, wherein the process state is saved in a checkpoint data structure.

16. The apparatus of claim 15, wherein the checkpoint data structure is a checkpoint file.

17. The apparatus of claim 15, wherein the checkpoint data structure includes a process descriptor for the process.

18. The apparatus of claim 17, wherein the process has control over at least one child process and the checkpoint data structure includes a process descriptor for each of the at least one child process.

19. The apparatus of claim 17, wherein the checkpoint data structure further includes at least one data type descriptor for at least one data type corresponding to the process.

Docket No. AUS9-2000-0277-US1

Sub A'

20. The apparatus of claim 19, wherein the checkpoint data structure further includes at least one instance descriptor for at least one instance of data corresponding to each of the at least one data type.

21. The apparatus of claim 20, wherein the checkpoint data structure further includes at least one data block corresponding to each of the at least one instance descriptor.

22. The apparatus of claim 13, wherein program further comprises the step of modifying at least one register or memory variable before resuming debugging from the retrieved process state.

23. The apparatus of claim 13, wherein the process state is saved when the program is in a stopped state.

24. The apparatus of claim 23, wherein the stopped state is at a breakpoint.

25. A computer program product embodied in a computer readable medium for debugging a process from a starting point, comprising:

instructions for initiating debugging of a process;

instructions for saving a process state in response to a first event to form a saved process state;

instructions for retrieving the saved process state in response to a second event; and

instructions for reinitiating debugging from the saved process state.

Add A²